

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Inventors:	Karen P. Parnell; Hans E. E. Kedefors; Atsushi Kaneko; Daniel Salzer; Jayant Kulkarni; Mark C. Hastings; Nikolai Tsepalov; David A. Murphy; Gloria Lorenzo-Jorge		
Assignee:	Siebel Systems, Inc.		
Title:	POLYLINGUAL SIMULTANEOUS SHIPPING OF SOFTWARE		
Application No.:	09/845,785	Filing Date:	April 30, 2001
Examiner:	Lamont M. Spooner	Group Art Unit:	2626
Docket No.:	OIC0044US	Confirmation No.:	4346

---

Austin, Texas  
December 7, 2009

Mail Stop: Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF**

Dear Sir:

This brief is submitted in support of the October 7, 2009 Notice of Appeal of the final rejection of claims 1-2, 5, 9, 16-19, 22, 26, 33-34, 38-43, and 45-62. Appellant notes that the Notice of Appeal was received by the USPTO on October 7, 2009, thereby giving Appellant a two-month period, ending December 7, 2009, to file this paper.

Please charge deposit account No. 502306 for the fee of \$540.00 associated with this appeal brief. Please charge this deposit account for any additional sums which may be required to be paid as part of this appeal.

**I. REAL PARTY IN INTEREST**

The real party in interest on this appeal is Siebel Systems, Inc., assignee of record. On January 31, 2006, Siebel Systems, Inc. was acquired by Oracle Corporation.

**II. RELATED APPEALS AND INTERFERENCES**

There are no interferences or other appeals related to this application.

**III. STATUS OF CLAIMS**

Claims 1-2, 5, 9, 16-19, 22, 26, 33-34, 38-43, and 45-62 are pending in the application.

Claims 1-2, 5, 9, 16-19, 22, 26, 33-34, 38-43, and 45-62 stand rejected. Appellant appeals the rejection of claims 1-2, 5, 9, 16-19, 22, 26, 33-34, 38-43, and 45-62.

**IV. STATUS OF AMENDMENTS**

No amendments were filed subsequent to the final rejection of July 7, 2009.

## **V. SUMMARY OF CLAIMED SUBJECT MATTER**

Appellant's independent claim 1 is directed to a method that includes developing a base version of a computer-implemented application in a base language (*see, for example, FIG. 1, block 12; p. 1, lines 20-21; p. 6, lines 7-10; p. 7, lines 9-10*), facilitating an internationalization of the base version of the application (*see, for example, p. 1, lines 21-23; p. 6, lines 7-10; p. 7, lines 10-12; FIG. 1, block 14*), and facilitating, using a processor, a localization of the base version of the application (*see, for example, p. 1, lines 21-23; p. 6, lines 7-10; p. 7, lines 12-13; p. 6, lines 12-19; FIG. 1, block 16*).

The base version of the application includes language dependant code and language independent code (*see, for example, p. 5, lines 2-4; p. 12, lines 19-21*). The developing the base version of the application includes developing a plurality of stages of the base version of the application (*see, for example, p. 8, line 10—p. 9, line 3*). The plurality of stages of the base version of the application includes a first stage (*see, for example, id.; code developed and processed in FIG. 1, blocks 20, 30, 40*), a second stage (*see, for example, id.; code developed and processed in FIG. 1, blocks 22, 33, 42*), and a third stage (*see, for example, id.; code developed and processed in FIG. 1, blocks 24, 34, 44*).

The internationalization of the base version of the application includes pseudo localization of the language dependent code of the base version of the application (*see, for example, p. 17, lines 19-22; p. 18, lines 1-22*). The internationalization of the base version of the application includes an internationalization of the second stage (*see, for example, FIG. 1, block 32; p. 7, lines 10-12*).

The localization includes generating a base glossary (*see, for example, p. 13, lines 15-19; p. 20, lines 11—p. 21, line 11; p. 22, line 8—p. 23, line 6*). The localization of the base version of the application includes a localization of the first stage (*see, for example, FIG. 1, block 40; p. 7, lines 10-12*).

The internationalization of the second stage (*e.g., block 32*) is performed concurrently with the developing of the third stage (*e.g., block 24*) (*see, for example, p. 6, lines 20-24, "The three processes 12, 14, and 16 proceed in a concurrent relationship, whereby, generally speaking, as a stage of each of the processes is completed, the product of that stage is directed to another process for further development, while at the same time, the next stage of the original*

*process is initiated”; see also, p. 9, lines 4-9, describing the internationalization of code concurrently with the development of a subsequent stage of base code).*

The localization of the first stage (*e.g., block 40*) is performed concurrently with the internationalization of the second stage (*e.g., block 32*) (*see, for example, p. 6, lines 20-24; p. 10, lines 3-6; original claim 1, describing concurrent internationalization and localization processes*). The method also includes storing a localized version of the first stage in a memory (*see, for example, p. 6, lines 12-19*).

Dependent claim 45 depends on independent claim 1 and further recites that the localization of the base version of the application includes a localization of the second stage (*e.g., block 42*) concurrently with an internationalization of the third stage (*e.g., block 34*) (*see, for example, p. 6, lines 20-24; p. 10, lines 3-6; original claim 1, describing concurrent internationalization and localization processes*).

Claims 2, 5, 9, 16-17, 38-43, 46-47, and 58-60 depend on independent claim 1.

Independent claim 18 is directed to an article of manufacture that includes a computer-readable storage medium encoded with instructions that, if executed by a processor will cause the processor to perform operations (*see, for example, p. 6, lines 12-19*). The operations include developing a base version of a computer-implemented application in a base language (*see, for example, FIG. 1, block 12; p. 1, lines 20-21; p. 6, lines 7-10; p. 7, lines 9-10*), facilitating an internationalization of the base version of the application (*see, for example, p. 1, lines 21-23; p. 6, lines 7-10; p. 7, lines 10-12; FIG. 1, block 14*), and facilitating a localization of the base version of the application (*see, for example, p. 1, lines 21-23; p. 6, lines 7-10; p. 7, lines 12-13; FIG. 1, block 16*).

The base version of the application includes language dependant code and language independent code (*see, for example, p. 5, lines 2-4; p. 12, lines 19-21*). The developing the base version of the application includes developing a plurality of stages of the base version of the application (*see, for example, p. 8, line 10—p. 9, line 3*). The plurality of stages of the base

version of the application includes a first stage (*see, for example, id.; code developed and processed in FIG. 1, blocks 20, 30, 40*), a second stage (*see, for example, id.; code developed and processed in FIG. 1, blocks 22, 33, 42*), and a third stage (*see, for example, id.; code developed and processed in FIG. 1, blocks 24, 34, 44*).

The internationalization of the base version of the application includes pseudo localization of the language dependent code of the base version of the application (*see, for example, p. 17, lines 19-22; p. 18, lines 1-22*). The internationalization of the base version of the application includes an internationalization of the second stage (*see, for example, FIG. 1, block 32; p. 7, lines 10-12*).

The localization includes generating a base glossary (*see, for example, p. 13, lines 15-19; p. 20, lines 11—p. 21, line 11; p. 22, line 8—p. 23, line 6*). The localization of the base version of the application includes a localization of the first stage (*see, for example, FIG. 1, block 40; p. 7, lines 10-12*).

The internationalization of the second stage (*e.g., block 32*) is performed concurrently with the developing of the third stage (*e.g., block 24*) (*see, for example, p. 6, lines 20-24, “The three processes 12, 14, and 16 proceed in a concurrent relationship, whereby, generally speaking, as a stage of each of the processes is completed, the product of that stage is directed to another process for further development, while at the same time, the next stage of the original process is initiated”; see also, p. 9, lines 4-9, describing the internationalization of code concurrently with the development of a subsequent stage of base code*).

The localization of the first stage (*e.g., block 40*) is performed concurrently with the internationalization of the second stage (*e.g., block 32*) (*see, for example, p. 6, lines 20-24; p. 10, lines 3-6; original claim 1, describing concurrent internationalization and localization processes*).

Dependent claim 48 depends on independent claim 18 and further recites that the localization of the base version of the application includes a localization of the second stage (*e.g., block 42*) concurrently with an internationalization of the third stage (*e.g., block 34*) (*see,*

*for example, p. 6, lines 20-24; p. 10, lines 3-6; original claim 1, describing concurrent internationalization and localization processes).*

Claims 19, 22, 26, 33, 49-50, and 56-57 depend on independent claim 18.

Independent claim 34 is directed to a server that includes a memory and a processor, coupled to the memory, that is configured to execute a set of instructions stored in the memory (*see, for example, p. 6, lines 12-19*). The instructions are configured to cause the processor to facilitate an internationalization of a base version of an application (*see, for example, FIG. 1, block 12; p. 1, lines 20-21; p. 6, lines 7-10; p. 7, lines 9-10*). The internationalization of the base version of the application includes pseudo localization of the language dependent code of the base version of the application (*see, for example, p. 17, lines 19-22; p. 18, lines 1-22*). The instructions are also configured to facilitate a localization of the base version of the application (*see, for example, p. 1, lines 21-23; p. 6, lines 7-10; p. 7, lines 12-13; FIG. 1, block 16*).

The base version of the application includes language dependant code and language independent code (*see, for example, p. 5, lines 2-4; p. 12, lines 19-21*). The base version of the application includes a first stage (*see, for example, id.; code developed and processed in FIG. 1, blocks 20, 30, 40*), a second stage (*see, for example, id.; code developed and processed in FIG. 1, blocks 22, 33, 42*), and a third stage (*see, for example, id.; code developed and processed in FIG. 1, blocks 24, 34, 44*).

The internationalization of the base version of the application includes an internationalization of the second stage among a plurality of stages of the base version of the application (*see, for example, FIG. 1, block 32; p. 7, lines 10-12*).

The localization includes generating a base glossary (*see, for example, p. 13, lines 15-19; p. 20, lines 11—p. 21, line 11; p. 22, line 8—p. 23, line 6*).

The localization of the base version of the application includes a localization of the first stage (*see, for example, FIG. 1, block 40; p. 7, lines 10-12*).

The internationalization of the second stage (*e.g., block 32*) is performed concurrently with the developing of the third stage (*e.g., block 24*) (*see, for example, p. 6, lines 20-24, "The*

*three processes 12, 14, and 16 proceed in a concurrent relationship, whereby, generally speaking, as a stage of each of the processes is completed, the product of that stage is directed to another process for further development, while at the same time, the next stage of the original process is initiated”; see also, p. 9, lines 4-9, describing the internationalization of code concurrently with the development of a subsequent stage of base code).*

The localization of the first stage (e.g., block 40) is performed concurrently with the internationalization of the second stage (e.g., block 32) (*see, for example, p. 6, lines 20-24; p. 10, lines 3-6; original claim 1, describing concurrent internationalization and localization processes*). The method also includes storing a localized version of the first stage in a memory (*see, for example, p. 6, lines 12-19*).

Dependent claim 51 depends on claim 34, and further recites that the localization of the base version of the application includes a localization of the second stage (e.g., block 42) concurrently with an internationalization of the third stage (e.g., block 34) (*see, for example, p. 6, lines 20-24; p. 10, lines 3-6; original claim 1, describing concurrent internationalization and localization processes*).

Claims 52-55 depend on independent claim 34.

Independent claim 61 is directed to a method that includes storing (e.g., block 20) a first set of language dependent code in a memory (*see, for example, p. 6, lines 12-19; FIG. 1, block 12; p. 8, line 10—p. 9, line 3; p. 1, lines 20-21; p. 6, lines 7-10; p. 7, lines 9-10*), storing (e.g., block 22) a second set of language dependent code in the memory (*see, for example, id.; FIG. 1*), storing (e.g., block 24) a third set of language dependent code in the memory (*see, for example, id.; FIG. 1, block 24*), and storing (e.g., block 20) a first set of language independent code in the memory (*see, for example, id; p. 12, lines 19-21*). The memory includes one or more data storage devices (*see, for example, p. 6, lines 12-19*).

The first set of language dependent code includes code for a user interface of a first development stage of a computer-implemented application (*see, for example, p. 12, lines 19-21*;

*FIG. 1, block 20*). The first set of language dependent code includes first content in a base language (*see, for example, p. 12, lines 9-16*). The first set of language independent code includes code for the first development stage of the computer-implemented application (*see, for example, p. 12, lines 19-21; FIG. 1, blocks 12, 20*).

The method also includes modifying (*e.g., block 30*) the first set of language dependent code, which includes generating an internationalized version of the first set of language dependent code (*see, for example, p. 1, lines 21-23; p. 6, lines 7-10; p. 7, lines 10-12; FIG. 1, blocks 14*). The internationalized version of the first set of language dependent code includes the first content in the base language and indicators of the first content in the base language (*see, for example, p. 17, lines 19-22; p. 18, lines 1-22*).

The method also includes modifying (*e.g., block 40*) the internationalized version of the first set of language dependent code (*see, for example, p. 1, lines 21-23; p. 6, lines 7-10; p. 7, lines 12-13; FIG. 1, blocks 16*). The modifying the internationalized version of the first set of language dependent code includes generating a plurality of target-language versions of the first set of language dependent code (*see, for example, p. 12, lines 10-16*). Each target-language version of the first set of language dependent code includes translations into a corresponding target language of the first content in the base language (*see, for example, id.*). The modifying the internationalized version of the first set of language dependent code includes generating a base glossary for each of the target languages (*see, for example, p. 13, lines 15-19; p. 20, lines 11—p. 21, line 11; p. 22, line 8—p. 23, line 6*).

The second set of language dependent code includes code for a user interface of a second development stage of the computer-implemented application (*see, for example, p. 12, lines 19-21; FIG. 1, block 22*). The second set of language dependent code includes second content in the base language (*see, for example, p. 12, lines 9-16*).

The method also includes modifying the second set of language dependent code, which includes generating (*e.g., block 34*) an internationalized version of the second set of language dependent code (*see, for example, p. 1, lines 21-23; p. 6, lines 7-10; p. 7, lines 10-12; FIG. 1, blocks 14*). The internationalized version of the second set of language dependent code includes the second content in the base language and indicators of the second content in the base language (*see, for example, p. 17, lines 19-22; p. 18, lines 1-22*).



The third set of language dependent code includes code for a user interface of a third development stage of the computer-implemented application (*see, for example, p. 12, lines 19-21; FIG. 1, block 24*). The third set of language dependent code includes third content in the base language (*see, for example, p. 12, lines 9-16*).

The storing the second set of language dependent code (*e.g., block 22*) is performed only after commencement of the modifying the first set of language dependent code (*e.g., block 30*) (*see, for example, p. 6, lines 20-23; see also FIG. 1 and accompanying discussion in p. 6, line 23—p. 8, line 9*).

The storing the third set of language dependent code (*e.g., block 24*) is performed only after commencement of the modifying the internationalized version of the first set of language dependent code (*e.g., block 40*) (*see, for example, p. 6, lines 20-23; see also FIG. 1 and accompanying discussion in p. 6, line 23—p. 8, line 9*).

Dependent claim 62 depends on independent claim 61 and further recites that the storing the third set of language dependent code is performed only after commencement of the modifying the second set of language dependent code (*see, for example, p. 6, lines 20-23; see also FIG. 1 and accompanying discussion in p. 6, line 23—p. 8, line 9*).

**VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

1. Are claims 61 and 62<sup>1</sup> patentable under 35 U.S.C. § 103(a) over U.S. Patent No. 5,416,903 issued to Malcolm (“**Malcolm**”)?
2. Are claims 1-2, 5, 9, 16-19, 22, 26, 33-34, 38-43, 45-54, 56, 58, and 60 patentable under § 103(a) over U.S. Patent No. 6,442,516 issued to Lee et al. (“**Lee**”) in view of U.S. Patent No. 6,425,123 issued to Rojas et al. (“**Rojas**”) and Malcolm?
3. Are claims 55, 57, and 59<sup>2</sup> patentable under § 103(a) over Lee in view of Rojas, Malcolm, and U.S. Patent No. 6,185,729 issued to Watanabe, et al. (“**Watanabe**”)?

---

<sup>1</sup> The Final Office Action appears to include a clerical error regarding the listing of claims that are rejected over Malcolm as a single reference. On p. 15, the Final Office Action indicates that these claims are claims 1, 2, 5, 9, 16-19, 22, 26, 33-34, 38-43, and 45-53. However, only claims 61 and 62 are subsequently discussed in the context of this single reference. Thus, Appellant understands this rejection to apply only to claims 61 and 62. If this understanding is in error, Appellant requests clarification.

<sup>2</sup> The Final Office Action appears to include a clerical error regarding the listing of claims that are rejected over the combination of Lee, Rojas, Malcolm, and Watanabe. On p. 14, the Final Office Action indicates that these claims are claims 1, 2, 5, 9, 16-19, 22, 26, 33-34, 38-43, and 45-53. However, only claims 55, 57, and 59 are subsequently discussed in the context of this combination. Thus, Appellant understands this rejection to apply only to claims 55, 57, and 59. If this understanding is in error, Appellant requests clarification.

## VII. ARGUMENT

### 1. Claim rejections over Malcolm.

Claims 61 and 62 stand rejected under 35 U.S.C. § 103(a) as purportedly being unpatentable over Malcolm. Appellant respectfully submits that the claims each include limitations that are absent from the cited passages of Malcolm in view of the knowledge available to a person having ordinary skill in the art.

Appellant's independent claim 61 includes:

storing a first set of language dependent code in a memory, wherein . . . the first set of language dependent code comprises code for a user interface of a first development stage of a computer-implemented application,  
 modifying the first set of language dependent code, wherein the modifying the first set of language dependent code comprises generating an internationalized version of the first set of language dependent code, . . .  
 modifying the internationalized version of the first set of language dependent code, . . .  
 storing a second set of language dependent code in the memory, wherein . . . the storing the second set of language dependent code is performed only after commencement of the modifying the first set of language dependent code;  
 modifying the second set of language dependent code, wherein  
     the modifying the second set of language dependent code comprises generating an internationalized version of the second set of language dependent code, . . .  
 storing a third set of language dependent code in the memory, wherein . . .  
     the storing the third set of language dependent code is performed only after commencement of the modifying the internationalized version of the first set of language dependent code.

(Emphasis added.)

### **A. The Final Office Action's cursory treatment of the temporal limitations in claim 61 warrants, by itself, a reversal of the rejection of independent claim 61.**

The Final Office Action correctly acknowledges that the temporal requirements of claim 61 (and numerous other limitations) are absent from Malcolm. See Final Office Action, p. 17 (last paragraph)—p. 19 (third paragraph).

Claim 61 relates to a development of a computer-related application. The development is described as being divided in two different ways. First, the code being developed includes at

least three sets of code: a first, a second, and a third set of language dependent code. These sets of code correspond to a first, a second, and a third development stage of the computer-related application. Second, the operations are delimited into various recited acts, e.g., storing sets of code, modifying sets of code, generating internationalized versions of sets of code, and generating target-language versions of code.

Claim 61 recites various explicit temporal relationships between different acts that are performed on the different sets of code. For example, one set of language dependent code (the second set) is stored “only after” a modification has commenced for another set of language dependent code (the first set). That is, the storing of the second set of language dependent code does not occur at any time before the modifying has commenced for the first set of language dependent code.

In addition, yet another set of language dependent code (the third set) is stored “only after” a modification has commenced for an internationalized version of the first set of language dependent code. That is, the storing of the third set of language dependent code does not occur at any time before the modifying has commenced for the internationalized version of first set of language dependent code.

With regard to these limitations, the Final Office Action proposes:

by definition of a first stage to second stage, it is obvious, that in subsequent processing, a first stage would commence before action on a subsequent stage.

Final Office Action, p. 19.

Here, the Final Office Action brushes aside detailed limitations that are explicitly recited in claim 61. The Final Office Action proposes instead that Appellant’s use of the labels “first” and “second”—in Appellant’s own claims—somehow cause Malcolm to teach that the above-noted temporal limitations.

The rejection is unfounded because it relies on the use of the labels “first” and “second” in Appellant’s claims as a purported teaching of the cited art. Appellant notes, and the Final Office Action appears to agree, that this teaching is actually absent from the cited passages of Malcolm. See Final Office Action, p. 17 (last paragraph)—p. 19 (third paragraph).

Moreover, even if Malcolm could be understood to teach that a “first” set of code is acted upon before a “second” set of code (a point which Appellant does not concede), that would not render obvious the specific and particular limitations of claim 61, such as:

- “storing a second set of language dependent code in the memory . . . only after commencement of the modifying the first set of language dependent code,” or
- “storing a third set of language dependent code in the memory, . . . only after commencement of the modifying the internationalized version of the first set of language dependent code,”

as recited in independent claim 61.

**B. The Final Office Action’s assessment of Malcolm’s “stages” cannot be equated with the sets of code in Appellant’s claim 61, further warranting reversal.**

In addition, the Final Office Action has erred in its interpretation of Malcolm’s “stages.” The Office Action appears to take the position that Malcolm’s “stages” are equivalent to Appellant’s first, second, and third sets of language dependent code. This characterization of Malcolm is erroneous. Malcolm’s one single use of the word “stages”—on which the pending rejection heavily relies—comes from an observation that “a product progresses through various stages prior to the end product.” See Malcolm, 10:16-21 (emphasis added). Where Malcolm uses this term, it relates to phases in the development of a product, and not to portions of the product. The Final Office Action thus errs by applying this concept from Malcolm to Appellant’s first, second, and third sets of language dependent code.

The Advisory Action proposes on p. 3 that Malcolm stages include a *translation process* and a *development cycle*. This observation, even if it were correct, would only support Appellant’s contention that a person having ordinary skill in the art would not equate Malcolm’s “stages” with the first, second, and third sets of *language dependent code* in claim 61.

A correct understanding of Malcolm’s stages could analogize these stages to operations that are performed as part of an engineering/software project. Thus, it could be argued that Malcolm’s stages are roughly akin with (but are not equivalent to) the recited operations of storing sets of code, modifying sets of code, generating internationalized versions of sets of code,

generating target-language versions, etc. in claim 61. With this understanding, the next steps in the analysis would be to assess whether Malcolm's "stages" have the temporal relationships recited in claim 61.

They do not. As discussed below, these recited relationships in claim 61 are non-obvious over the teachings of Malcolm for at least two separate reasons. First, the interchangeability of Malcolm's "stages," as described in the Final Office Action, is contrary to the limitations in claim 61. Second, as also discussed below, Malcolm's mention of activities done "in parallel" does not render obvious the particular temporal relationships in Appellant's claim 61.

**B1. The Final Office Action's assessment of interchangeable stages is contrary to the claim 61's temporal relationships between distinct operations on different sets of code.**

The Final Office Action also acknowledges that Malcolm fails to individually address various different sets of code. See Final Office Action, p. 17 (last paragraph)—p. 19 (third paragraph). Rather, the Final Office Action conjectures that the cited operations in Malcolm are for a "first stage," and proposes that:

all the elements for [Malcolm's] first stage . . . may be applied to each stage of the development.

See, Final Office Action, p. 19.

As discussed above, the Office Action appears to erroneously take the position that Malcolm's "stages" are equivalent to Appellant's first, second, and third sets of language dependent code. In addition, the Office Action asserts that whatever actions are performed on one of these "stages" can be applied to the others as well. The Final Office Action thus appears to view Malcolm's operations as interchangeable with regard to different sets of code, without particular considerations regarding the different code. This stance is reiterated in the Advisory Action, which proposes that in Malcolm "the stages may be divided in any manner." See, Advisory Action, p. 4.

Even if these characterizations of Malcolm were appropriate (a point which Appellant certainly does not concede), they would fail to support the rejection of claim 61. At best, these

characterizations of Malcolm would make clear that Malcolm fails to distinguish between operations on one set of code and corresponding operations on another set of code.

That lack of distinction is counter to the limitations of claim 61. Claim 61 makes explicitly clear that the first, second, and third sets of language dependent code are addressed distinctly—and not simply in “any manner,” as the Advisory Action appears to find within the teachings of Malcolm. In particular, as explained above, certain operations on various sets of language dependent code (e.g., the storing the third set of language dependent code) must be performed with a particular consideration to operations on other sets of language dependent code (e.g., only after commencement of the modifying the internationalized version of the first set of language dependent code). The Final Office Action’s proposition—that Malcolm’s “stages” can be treated interchangeably—further emphasizes the shortcomings of Malcolm, which fails to teach the particular temporal order of actions that is recited in Appellant’s claim 61. At least for these reasons, independent claim 61 is allowable under § 103(a) over the cited passages of Malcolm.

**B2. *Malcolm’s mention of a general engineering principle does not render obvious all uses of that principle.***

Malcolm states:

As a further aid in the translation process, it has been found to be extremely useful to track and log changes made during the development of the initial panels. Such changes are common in a typical engineering/software development cycle, when a product progresses through various stages prior to the end product. For example, testing may discover errors in the program, the user interface may be objected to by a human-factors specialist, etc. Yet, in order to decrease development time of products (or ‘time to market’), numerous activities must be done in parallel to reduce the overall time requirements. Therefore, a set of screen panels for a given application may need to be sent to a translation center before the final program code is completed. This is desirable in that translation centers can add significant time delays in translating a product, and by getting this phase going early on, prior to final code delivery, the overall development time can potentially be reduced. However, when the translation centers begin work on a pre-release version of code, the potential exists that what is desired in the final product may have different screen

panels, requiring a different language dependent file for translation. There is no convenient method for indicating how a subsequent language dependent file differs from an earlier received version.

Malcolm, 10:16-56 (emphasis added).

This passage mentions the “stages” on which the Final Office Action relies for the pending rejection of claim 61. Malcolm teaches that the development of an engineering/software product typically progresses through various stages prior to the end product. See Malcolm, 10:16-21. This passage further recognizes that “numerous activities must be done in parallel” to reduce the overall time requirements in development.

However, different development environments need to carefully and judiciously select which activities need to be done in parallel with which other activities, if the result is to be an effective reduction in the total development time. In the particular case of Malcolm, the reference goes on to describe one particular selection that is helpful in that environment: the translation of screen panels can begin before the final code is completed. See Malcolm, 10:27-29.

The Final Office Action appears to take the position that this one example—Malcolm’s translation of screen panels—portends all possible evaluations and engineering decisions that can be made in selecting parallel operations. Appellant respectfully disagrees. A general recognition that a particular group of processes can be performed in parallel (as in Malcolm) does not make it a trivial or obvious task to find the relevant solutions in other situations. In particular, this general observation from Malcolm fails to render obvious the particular requirements of “storing a second set of language dependent code in the memory . . . only after commencement of the modifying the first set of language dependent code,” or “storing a third set of language dependent code in the memory, . . . only after commencement of the modifying the internationalized version of the first set of language dependent code,” as recited in claim 61.

At least for these reasons, independent claim 61 is allowable under § 103(a). At least for similar reasons, claim 62 is also allowable under § 103(a), being dependent on an allowable base claim.



**C. Dependent claim 62 additionally distinguishes over Malcolm.**

Claim 62 depends on claim 61 and additionally recites:

the storing the third set of language dependent code is performed only after commencement of the modifying the second set of language dependent code.

(Emphasis added.) This additional temporal limitation is also absent from the cited passages of Malcolm.

The Final Office Action is somewhat ambiguous regarding the purported teachings of Malcolm with regard to this limitation of claim 62. On p. 22, the Final Office Action appears to assert that “all of the components are present” in Malcolm, but does not point to any particular teaching of Malcolm as disclosing the above limitation of claim 62. Appellant certainly disagrees with the assertion that “all of the components are present,” since the Final Office Action fails to even propose where the above-noted limitation could be found in Malcolm, or within the knowledge available to a person having ordinary skill in the art.

To the contrary, the Final Office Action acknowledges on p. 21 that “Malcolm does not explicitly teach the above” limitation of claim 62. Appellant agrees with this assessment, and respectfully submits that the Final Office Action fails to adequately explain why the rejection is justified in view of this explicit shortcoming of Malcolm with regard to claim 62.

A rejection under § 103(a) may establish, among others, that “all the claimed elements were known in the prior art.” See, MPEP § 2143.02 (citing *KSR International Co. v. Teleflex Inc.*, 550 U.S. \_\_\_, \_\_\_, 82 USPQ2d 1385, 1395 (2007)). The pending rejection of claim 62 attempts to meet this standard, but falls short because of undisputed infirmity of Malcolm. Moreover, Appellant does not find this limitation in Malcolm. At least for this reason, dependent claim 62 is additionally allowable under § 103(a).

**2. Claim rejections over Lee in view of Rojas and Malcolm.**

Claims 1-2, 5, 9, 16-19, 22, 26, 33-34, 38-43, 45-54, 56, 58, and 60 stand rejected under § 103(a) as purportedly being unpatentable over Lee in view of Rojas and Malcolm. Appellant respectfully submits that the claims each include limitations that are absent from the cited passages of Lee, Rojas, or Malcolm, whether taken individually or in combination, in view of the knowledge available to a person having ordinary skill in the art.

**A. Independent claims 1, 18, and 34 distinguish over the cited passages.**

For example, independent claim 1 includes developing a first stage, a second stage, and a third stage of the base version of an application; facilitating an internationalization of the base version of the application; and facilitating a localization of the base version of the application. Claim 1 further recites that various operations include specific temporal relationships:

the internationalization of the second stage is performed concurrently with the developing of the third stage; and . . .

the localization of the first stage is performed concurrently with the internationalization of the second stage.

Appellant respectfully submits that, among others, these limitations regarding the concurrent timing of particular activities are not disclosed or fairly suggested in the cited passages of the references.

With regard to these limitations, the Office Action hypothesizes that:

The development of [Malcolm's] product could be done simultaneously, concurrently, and in parallel.

See, Final Office Action, p. 3. This assertion (actually, merely a speculation) regarding Malcolm is not supported by the passages actually cited.

With regard to these limitations, the Final Office Action initially cites elements of Lee and Malcolm. See Final Office Action, pp. 2-3. The Final Office Action correctly acknowledges, however, that Lee (and Rojas as well) fail to disclose the temporal limitations that various acts are performed “concurrently.” See Final Office Action, p. 6, second paragraph—

p. 7, first paragraph. Finally, the Final Office Action turns on p. 7 to the mention in Malcolm of an example in which:

“numerous activities must be done in parallel.”

See, Malcolm, 10:16-35. This appears to be the only teaching of the cited art on which the Final Office Action relies with regard to the above-noted limitations.

The rejection of claim 1 thus ultimately rests on the same single passage from Malcolm that was discussed above with regard to Appellant’s independent claim 61. The above arguments regarding differences between Malcolm and the temporal limitations in claim 61 apply with equal force to the rejection of independent claim 1. In particular, Malcolm’s recognition that certain processes can be performed in parallel in Malcolm’s example does not render obvious the particular solutions that would be relevant in other situations. In particular, the example from Malcolm fails to render obvious the particular limitations that “the internationalization of the second stage is performed concurrently with the developing of the third stage,” or that “the localization of the first stage is performed concurrently with the internationalization of the second stage” as recited in claim 1.

The Final Office Action acknowledges that these limitations are absent from Lee and Rojas. See Final Office Action, p. 6. As discussed above, the Final Office Action also does not find these specific limitations in Malcolm, but resorts instead to a mention of activities “done in parallel.” See Final Office Action, p. 7.

The Advisory Action mentions Lee, Rojas, and Malcolm, but does not propose that any particular teachings of those references disclose the above-noted limitations. Moreover, Appellant does not find these limitations in the cited passages of Lee, Rojas, or Malcolm, whether taken individually or in combination in view of the knowledge available to a person having ordinary skill in the art. These limitations are therefore absent from the cited passages. At least for these reasons, independent claim 1 is allowable under § 103(a).

At least for similar reasons, independent claims 18 and 34 are also allowable under § 103(a). Claims 2, 5, 9, 16-17, 19, 22, 26, 33, 38-43, 45-54, 56, 58, and 60 are similarly allowable under § 103(a), being dependent on allowable base claims.

**B. Dependent claims 45, 48, and 51 additionally distinguish over the cited passages.**

Claim 45 depends on claim 1 and additionally recites that:

the localization of the base version of the application comprises a  
localization of the second stage concurrently with an  
internationalization of the third stage.

(Emphasis added.) This additional temporal limitation is also absent from the cited passages.

The Final Office Action turns, on p. 11, to the following two features in support of the rejection of claim 45:

- Lee’s “concurrent builds” performed on national language support build releases. See Lee, 3:56-61.

Lee teaches that these builds can be performed “at any time”—which actually teaches against the specific temporal limitation recited in claim 45. Moreover, the cited passage is unclear regarding the meaning of “concurrent.” In particular, the cited passage does not appear to teach that the Lee’s national language support build releases (which the Final Office Action appears to view as a localization) are concurrent with any internationalization operations.

- “Malcolm teaches stages.” See, Final Office Action, p. 11; but note Appellant’s above discussion of the inaptness of Malcolm’s “stages.”

The Final Office Action proposes that Lee’s “concurrent builds” should somehow be combined with Malcolm’s “stages” to achieve the above-noted limitation of claim 45. But even if the cited references actually taught the above two points as proposed by the Final Office Action (a proposition with which Appellant does not agree), the Final Office Action nonetheless fails to explain how these teachings should be combined to achieve Appellant’s claim 45. For example, the Final Office Action does not attempt to explain why the resulting combination would particularly perform a localization of a second stage concurrently with an internationalization of a third stage.

As discussed above, a rejection under § 103(a) may establish, among others, that “all the claimed elements were known in the prior art.” See, MPEP § 2143.02 (citing *KSR International*

*v. Teleflex*, 550 U.S. \_\_\_, \_\_\_, 82 USPQ2d 1385, 1395). The pending rejection of claim 45 attempts to meet this standard, but falls short because the cited passages fail, even in combination in view of the knowledge available to a person having ordinary skill in the art, to disclose the particular temporal relationship presented in claim 45. At least for this reason, dependent claim 45 is additionally allowable under § 103(a). At least for similar reasons, dependent claims 48 and 51 are also additionally allowable under § 103(a).

**3. Claim rejections over Lee in view of Rojas, Malcolm, and Watanabe.**

Claims 55, 57, and 59 stand rejected under § 103(a) as purportedly being unpatentable over Lee in view of Rojas, Malcolm, and Watanabe. Claims 55, 57, and 59 depend, respectively, on independent claims 34, 18, and 1, discussed above. The Final Office Action does not propose that the above-noted infirmities of Lee, Rojas, and Malcolm are remedied in Watanabe. Thus, the above observations regarding the patentability of independent claims 1, 18, and 34 apply with full force to the rejections of claims 55, 57, and 59. At least for these reasons, claims 55, 57, and 59 are also allowable under § 103(a).

**Conclusion**

Appellant respectfully submits that claims 1-2, 5, 9, 16-19, 22, 26, 33-34, 38-43, and 45-62 are allowable over the cited references for at least the above-stated reasons. Appellant respectfully requests that the Board reverse the rejections of these claims.

I hereby certify that this correspondence is being submitted to the U.S. Patent and Trademark Office in accordance with 37 C.F.R. § 1.8 on December 7, 2009 (CT) by being (a) transmitted via the USPTO's electronic filing system; or (b) transmitted by facsimile to 571-273-8300; or (c) deposited with the U.S. Postal Service as First Class Mail in an envelope with sufficient postage addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, Virginia, 22313-1450.

/ Cyrus F. Bharucha /  
Cyrus F. Bharucha

December 7, 2009  
Date

Respectfully submitted,

/ Cyrus F. Bharucha /

Cyrus F. Bharucha  
Attorney for Appellant  
Reg. No. 42,324  
512-439-5097  
512-439-5099 (fax)

**VIII. CLAIMS APPENDIX**

1. A method comprising:  
developing a base version of a computer-implemented application in a base language,  
wherein  
the base version of the application comprises language dependant code and  
language independent code,  
the developing the base version of the application comprises developing a  
plurality of stages of the base version of the application, and  
the plurality of stages of the base version of the application comprises a first  
stage, a second stage, and a third stage;  
facilitating an internationalization of the base version of the application, wherein  
the internationalization of the base version of the application comprises pseudo  
localization of the language dependent code of the base version of the  
application,  
the internationalization of the base version of the application comprises an  
internationalization of the second stage, and  
the internationalization of the second stage is performed concurrently with the  
developing of the third stage; and  
facilitating, using a processor, a localization of the base version of the application,  
wherein  
the localization comprises generating a base glossary, and  
the localization of the base version of the application comprises a localization of  
the first stage, and  
the localization of the first stage is performed concurrently with the  
internationalization of the second stage; and  
storing a localized version of the first stage in a memory.

2. The method of claim 1 wherein developing the base version of the application comprises:  
identifying all language dependent user interface code in the base version of the application; and  
creating a source code structure for the application wherein the language dependent user interface code is maintained separately from non user interface code.
- 3-4. (Canceled)
5. The method of claim 1 wherein the base language is English.
- 6-8. (Canceled)
9. The method of claim 1 wherein pseudo localization includes adding a prefix to each translatable string in the application.
- 10-15. (Canceled)
16. The method of claim 1 wherein the at least one language different from the base language is selected from the group consisting of: German, Spanish, French, Japanese, Danish, Dutch, Italian, Portuguese, Swedish, Chinese, Korean, Czech, Finnish, Greek, and Hebrew.
17. The method of claim 1 wherein the application comprises a front end, a middle, and a data model, wherein the front end comprises user interface code developed in a base language and the middle comprises non user interface code developed in a programming language.



18. An article of manufacture, comprising a computer-readable storage medium encoded with instructions that, if executed by a processor, will cause the processor to perform operations comprising:

developing a base version of a computer-implemented application in a base language,  
wherein

the base version of the application comprises language dependant code and  
language independent code,

the developing the base version of the application comprises developing a  
plurality of stages of the base version of the application, and

the plurality of stages of the base version of the application comprises a first  
stage, a second stage, and a third stage;

facilitating an internationalization of the base version of the application, wherein

the internationalization of the base version of the application comprises pseudo  
localization of the language dependent code of the base version of the  
application,

the internationalization of the base version of the application comprises an  
internationalization of the second stage, and

the internationalization of the second stage is performed concurrently with the  
developing of the third stage; and

facilitating a localization of the base version of the application, wherein

the localization comprises generating a base glossary,

the localization of the base version of the application comprises a localization of  
the first stage, and

the localization of the first stage is performed concurrently with the  
internationalization of the second stage.

19. The article of manufacture of claim 18 wherein developing the base version of the application comprises:

identifying all user interface code in the base version of the application as language dependant code; and

creating a source code structure for the application wherein the user interface code is maintained separately from non user interface code.

20-21. (Canceled)

22. The article of manufacture of claim 18 wherein the base language is English.

23-25. (Canceled)

26. The article of manufacture of claim 18 wherein pseudo localization includes adding a prefix to each translatable string in the application.

27-32. (Canceled)

33. The article of manufacture of claim 18 wherein the at least one language different from the base language is selected from the group consisting of: German, Spanish, French, Japanese, Danish, Dutch, Italian, Portuguese, Swedish, Chinese, Korean, Czech, Finnish, Greek, and Hebrew.

34. A server comprising:

a memory; and

a processor, coupled to the memory, and configured to execute a set of instructions stored in the memory, wherein

the instructions are configured to cause the processor to facilitate an internationalization of a base version of an application,

the base version of the application comprises language dependant code and language independent code,

the base version of the application comprises a first stage, a second stage, and a third stage,

the internationalization of the base version of the application comprises pseudo localization of the language dependent code of the base version of the application,

the internationalization of the base version of the application comprises an internationalization of the second stage among a plurality of stages of the base version of the application,

the internationalization of the second stage is performed concurrently with the developing of the third stage,

the instructions are configured to cause the processor to facilitate a localization of the base version of the application,

the localization comprises generating a base glossary,

the localization of the base version of the application comprises a localization of the first stage, and

the localization of the first stage is performed concurrently with the internationalization of the second stage.

35-37. (Canceled)

38. The method of claim 1 wherein a first portion of the language dependent code is stored in a master repository and a second portion of the language dependent code is stored in resource files.

39. The method of claim 1 wherein the internationalization of the base version of the application further comprises identifying defects in a previous version of the application.

40. The method of claim 9 wherein the pseudo localization further comprises altering locale-specific settings in an operating environment.
41. The method of claim 40 wherein the locale-specific settings comprise at least one of a date, a time, a number, a currency format and a hard-coded reference to a translation.
42. The method of claim 9 wherein the pseudo localization further comprises identifying hard-coded strings in the application by simulating localization of the application.
43. The method of claim 1 wherein generating the base glossary comprises creating a list of base language strings.
44. (Canceled)
45. The method of claim 1 wherein the localization of the base version of the application comprises a localization of the second stage concurrently with an internationalization of the third stage.
46. The method of claim 1 wherein the internationalization of the base version of the application comprises adapting the base version of the application to be capable of being localized in a variety of locales.
47. The method of claim 1 wherein the base glossary comprises a glossary for the language dependent code, translated into at least one language different from the base language.
48. The article of manufacture of claim 18 wherein the localization of the base version of the application comprises a localization of the second stage concurrently with an internationalization of the third stage.

49. The article of manufacture of claim 18 wherein the internationalization of the base version of the application comprises adapting the base version of the application to be capable of being localized in a variety of locales.

50. The article of manufacture of claim 18 wherein the base glossary comprises a glossary for the language dependent code, translated into at least one language different from the base language.

51. The server of claim 34 wherein the localization of the base version of the application comprises a localization of the second stage concurrently with an internationalization of the third stage.

52. The server of claim 34 wherein the internationalization of the base version of the application comprises adapting the base version of the application to be capable of being localized in a variety of locales.

53. The server of claim 34 wherein the base glossary comprises a glossary for the language dependent code, translated into at least one language different from the base language.

54. The server of claim 34 wherein the language dependant code of the base version of the application is stored separately from the language independent code of the base version of the application.

55. The server of claim 34 wherein the instructions are further configured to cause the processor to:

modify the base version of the application, in response to at least one of the internationalization of the base version of the application or the localization of the base version of the application.

56. The article of manufacture of claim 18 wherein the language dependant code of the base version of the application is maintained separately from the language independent code of the base version of the application.
57. The article of manufacture of claim 18 wherein the operations further comprise:  
modifying the base version of the application, wherein the modifying is performed in response to at least one of: the internationalization of the base version of the application or the localization of the base version of the application.
58. The method of claim 1 wherein the language dependant code of the base version of the application is stored separately from the language independent code of the base version of the application.
59. The method of claim 1 further comprising:  
modifying the base version of the application, wherein the modifying is performed in response to at least one of: the internationalization of the base version of the application or the localization of the base version of the application.
60. The method of claim 1 further comprising:  
providing the localized version of the first stage for testing using an input/output device.
61. A method comprising:  
storing a first set of language dependent code in a memory, wherein  
the memory comprises one or more data storage devices,  
the first set of language dependent code comprises code for a user interface of a first development stage of a computer-implemented application, and  
the first set of language dependent code comprises first content in a base language,

storing first set of language independent code in the memory, wherein

the first set of language independent code comprises code for the first development stage of the computer-implemented application,

modifying the first set of language dependent code, wherein

the modifying the first set of language dependent code comprises generating an internationalized version of the first set of language dependent code,

the internationalized version of the first set of language dependent code comprises the first content in the base language, and indicators of the first content in the base language;

modifying the internationalized version of the first set of language dependent code, wherein

the modifying the internationalized version of the first set of language dependent code comprises generating a plurality of target-language versions of the first set of language dependent code,

each target-language version of the first set of language dependent code comprises translations into a corresponding target language of the first content in the base language, and

the modifying the internationalized version of the first set of language dependent code comprises generating a base glossary for each of the target languages;

storing a second set of language dependent code in the memory, wherein

the second set of language dependent code comprises code for a user interface of a second development stage of the computer-implemented application,

the second set of language dependent code comprises second content in the base language, and

the storing the second set of language dependent code is performed only after commencement of the modifying the first set of language dependent code;

modifying the second set of language dependent code, wherein

the modifying the second set of language dependent code comprises generating an internationalized version of the second set of language dependent code,

the internationalized version of the second set of language dependent code comprises

the second content in the base language, and

indicators of the second content in the base language;

storing a third set of language dependent code in the memory, wherein

the third set of language dependent code comprises code for a user interface of a third development stage of the computer-implemented application,

the third set of language dependent code comprises third content in the base language, and

the storing the third set of language dependent code is performed only after commencement of the modifying the internationalized version of the first set of language dependent code.

62. The method of claim 61, wherein:

the storing the third set of language dependent code is performed only after commencement of the modifying the second set of language dependent code.



**IX. EVIDENCE APPENDIX**

None

**X. RELATED PROCEEDINGS APPENDIX**

None